

E.T.S. de Ingeniería Industrial, Informática y de Telecomunicación

Diseño e implementación de un plugin WordPress para la creación de catálogos de productos compatibles con WooCommerce



Grado en Ingeniería Informática

Trabajo Fin de Grado

Julen Sanzol Tous

Iñigo Fermin Ezcurdia Aguirre

Pamplona, 3 de junio de 2021

Resumen

El producto desarrollado como trabajo de fin de grado consiste en un plugin de WordPress que permite crear un catálogo de productos. Este catálogo está configurado de tal manera que todos los productos creados y todos los datos introducidos se transfieren de forma automática al plugin de Woocommerce en el caso de querer crear una tienda online a partir de un catálogo ya existente. Este plugin de WordPress es una plantilla creada para la empresa TTANDEM a partir de la cual podrán desarrollar proyectos a medida para futuros clientes. Las funcionalidades implementadas son, la creación de productos, creación de índices de agrupamiento de por categorías y etiquetas, creación de atributos con la misma estructura que los atributos de Woocommerce, salida de datos a través de la API e implementación de una página de configuración de enlaces y nombres.

Palabras clave

WordPress, TTANDEM, Woocommerce, eCommerce, web, catálogo, plugin.

Glosario

API: Application Programming Interface, es decir, una interfaz de programación de aplicaciones. Permite que servicios y productos se comuniquen entre sí sin tener que saber cómo están implementados.

API REST: Es una api ajustada a los límites de la arquitectura REST, que permite interactuar con los servicios web de RESTful.

Atributo: Cualidades de un sujeto.

Base de datos: Definido mediante los acrónimos DB o BD, es una entidad donde almacenar datos de manera estructurada.

Callback: En programación, es una devolución de la llamada.

Catálogo: Medio impreso o virtual donde las empresas muestran su inventario de productos.

Codex de WordPress: Colección de normas, código y explicaciones sobre el funcionamiento de WordPress.

Custom Post Type: Entradas con funcionalidad personalizada en WordPress.

eCommerce: Conocido también como comercio electrónico. Consiste en la compraventa de servicios o productos a través de internet.

Endpoint: También llamada ruta de acceso. Son las URLs que usa la API.

Etiqueta: Una etiqueta de un producto es un sustantivo con el que catalogar la función o ámbito de uso del objeto.

Feedback: Ofrecer información a una persona sobre un resultado.

Hook: También conocido como gancho. Punto concreto dentro de WordPress que permite modificar su comportamiento o añadir funcionalidades.

Metabox: Herramienta que permite crear campos de personalización para cualquier entrada de WordPress.

Plugin: Fragmento de código que añade o modifica funcionalidades en WordPress.

Taxonomía: Clasificación u ordenación de objetos por grupos de características comunes.

Tema: Diseño completo de un sitio web gestionado por WordPress.

Woocommerce: Plugin de comercio electrónico de código abierto de WordPress.

WordPress: Sistema de gestión de contenidos enfocado a crear páginas web.

Contenido

| | |
|---|----|
| Resumen | 2 |
| Palabras clave | 2 |
| Glosario | 2 |
| Introducción | 6 |
| Objetivos | 7 |
| Contexto tecnológico | 8 |
| Metodología empleada | 9 |
| Análisis y diseño | 11 |
| Configuración del entorno de trabajo | 11 |
| WordPress | 12 |
| Plugins | 12 |
| Temas | 14 |
| Funciones de WordPress | 17 |
| Base de datos de WordPress | 20 |
| API de WordPress | 20 |
| Estudio de productos anteriores de TTANDEM | 21 |
| Plugins de terceros | 21 |
| Gutenberg | 21 |
| CMB2 | 22 |
| Estudio de compatibilidad de Woocommerce | 22 |
| Primer periodo de investigación | 23 |
| Segundo periodo de investigación | 24 |
| Desarrollo del proyecto | 26 |
| Fase 1: Creación del Custom Post Type con dos taxonomías | 26 |
| Fase 2: Página de configuración | 27 |
| Fase 3: Tema estético o plantilla | 28 |
| Fase 4: REST API | 30 |
| Fase 5: Atributos de Woocommerce | 31 |
| Fase de pruebas y resultados | 34 |
| Pruebas del producto y taxonomías de categorías y etiquetas | 34 |
| Pruebas de la página de configuración CMB2 | 36 |
| Pruebas del tema | 37 |
| Pruebas del endpoint | 37 |
| Pruebas de funcionalidad de atributos | 38 |

| | |
|---------------------|----|
| Conclusiones | 42 |
| Cierre del proyecto | 43 |
| Bibliografía | 45 |
| ANEXO 1 | 46 |
| ANEXO 2 | 52 |

Introducción

En este documento se explicará de forma detallada el desarrollo del producto que se ha creado en la empresa TTANDEM Digital Studio durante el transcurso del semestre de primavera en las prácticas de empresa.

TTANDEM es una empresa situada en el pueblo navarro Artica que se dedica a acompañar a empresas y profesionales en el desarrollo del negocio, enfocándose en la digitalización y estrategia empresarial. Está formada por seis socios que fueron quienes crearon la empresa en el 2013 y actualmente cuentan con colaboradores en distintos países como Estados Unidos.

En varios proyectos de digitalización de empresas, TTANDEM ha desarrollado catálogos de productos web a medida donde los clientes pudieran mostrar sus productos. En diversas ocasiones dichos clientes al cabo de un tiempo les pedían convertir ese catálogo creado anteriormente en una tienda online, lo cual a TTANDEM le suponía crear la tienda con una herramienta como Woocommerce y viéndose obligados a pasar todos los datos del catálogo a la nueva tienda manualmente. Este proyecto pretende atajar este problema ofreciendo una alternativa automatizada.

La idea inicial del proyecto es generar un plugin de WordPress que elimine la necesidad de traspasar los datos de los catálogos a mano a la hora de crear una tienda online con el plugin de Woocommerce. Para esto se plantea la posibilidad de que los productos creados en el nuevo plugin sean compatibles con los productos de Woocommerce o que de algún modo tengan una estructura similar. Esto permitirá que, al tener que crear una tienda online para un cliente al que anteriormente se le ha creado un catálogo, al iniciar la nueva tienda todos los datos del catálogo estén disponibles de forma automática.

Se advierte al lector del presente documento de que la empresa TTANDEM solicita que cierta información permanezca con carácter confidencial. Por ello, ciertos detalles de la investigación y del funcionamiento interno de la herramienta implementada han sido omitidos intencionalmente de este documento. Ejemplos de esto son la codificación de ciertas funciones, de palabras clave que utiliza Woocommerce, funcionamiento interno e información de las tablas de la base de datos. Se invita al lector a contactar con el autor y/o con TTANDEM para obtener más información al respecto si así lo desea.

Objetivos

La idea inicial del proyecto es crear un plugin de Wordpress para la empresa TTANDEM que permita crear un catálogo de productos. Además, el plugin de Woocommerce debe ser capaz de reconocer ese catálogo como propio para transferir los datos de forma automática entre los dos plugins. Las funcionalidades que debe tener este plugin deben permitir crear una plantilla de un catálogo, para que TTANDEM pueda utilizar esa plantilla para desarrollar un catálogo personalizado para futuros clientes.

Cómo se desconocen los requisitos que debe cumplir esta plantilla de catálogo y la cantidad de propiedades compatibles con Woocommerce que será posible implementar, se especifican los siguientes objetivos para la creación del plugin:

- Se realizará un estudio de distintos proyectos de catálogos desarrollados por TTANDEM para recopilar necesidades comunes y generar los requisitos de creación de la plantilla de catálogo.
- Se investigará el plugin de Woocommerce para comprobar cuáles de los requisitos del apartado anterior son posibles de implementar para que sean compatibles.
- En cuanto a aspectos técnicos y de calidad, la plantilla creada deberá tener funcionalidad que permitan modificar aspectos básicos del catálogo para facilitar su futura personalización.
- Una vez finalizado, el plugin deberá ser totalmente funcional sin depender del plugin de Woocommerce. Woocommerce se usará solo durante el desarrollo y al momento de convertir el catálogo en una tienda online.
- Se diseñará e implementará un tema estético básico de WordPress que sirva para mostrar las funcionalidades que añade el plugin.
- Deberá ser compatible con WPML (multiidioma).
- Deberá disponer de una salida de datos en la API de WordPress que retorne productos.
- Una vez finalizado deberá someterse a pruebas de funcionamiento y validación.

Contexto tecnológico

Antes de iniciar el desarrollo del producto, se establecen con TTANDEM las herramientas y lenguajes a utilizar.

- ❑ Git: Se crea un repositorio en GitHub con el fin de tener un control de versiones y un medio para compartir el proyecto.
- ❑ OneNote para Windows 10: Herramienta usada para compartir documentos e información relevante para investigar.
- ❑ Skype: Plataforma de llamadas online para efectuar las reuniones de equipo.
- ❑ Twist: Plataforma de mensajería instantánea para la comunicación del día a día.
- ❑ Asana: Es una herramienta web de gestión de flujos de trabajo, se usa para asignar y controlar actividades del proyecto.
- ❑ Sublime Text 3 con el paquete Emmet: Sublime Text 3 es un editor de texto enfocado a la programación en muchos lenguajes, y el paquete instalado Emmet añade atajos de texto y ayudas para programación HTML y CSS, lo que permite una escritura más rápida.
- ❑ Wamp 3.2.3: Es un conjunto de software que incluye Apache, MySQL y PHP. Además, actúa como servidor virtual para poder probar WordPress sin necesidad de contratar ningún hosting.
- ❑ WordPress: Es un sistema de gestión de contenidos enfocado a la creación de distintos tipos de sitios web. Como es la herramienta principal que se ha usado durante el desarrollo del proyecto, se explicará más a fondo posteriormente.

Metodología empleada

La realización del proyecto ha seguido una metodología mixta de TTANDEM basada en los fundamentos de Getting Things Done (GTD) y la metodología Scrum.

GTD es un método de gestión de actividades que se enfoca en guardar las tareas pendientes en un lugar específico para que la persona pueda liberar su mente. En este caso, las tareas pendientes se almacenan en la herramienta web Asana y se van desarrollando una a una por separado, una vez completada una tarea se inicia la siguiente.

En cuanto al feedback y reuniones, durante el proceso de desarrollo se usa un calendario basado en la metodología Scrum donde cada día se plantea que se va a realizar y además una vez a la semana hay una reunión general para corrección de errores e informar de adelantos.

Al comienzo del proceso se definieron y estimaron temporalmente distintas tareas y cargas de trabajo. En el Diagrama 1 se exponen las distintas cargas de trabajo junto a su desvío temporal respecto a la estimación inicial.

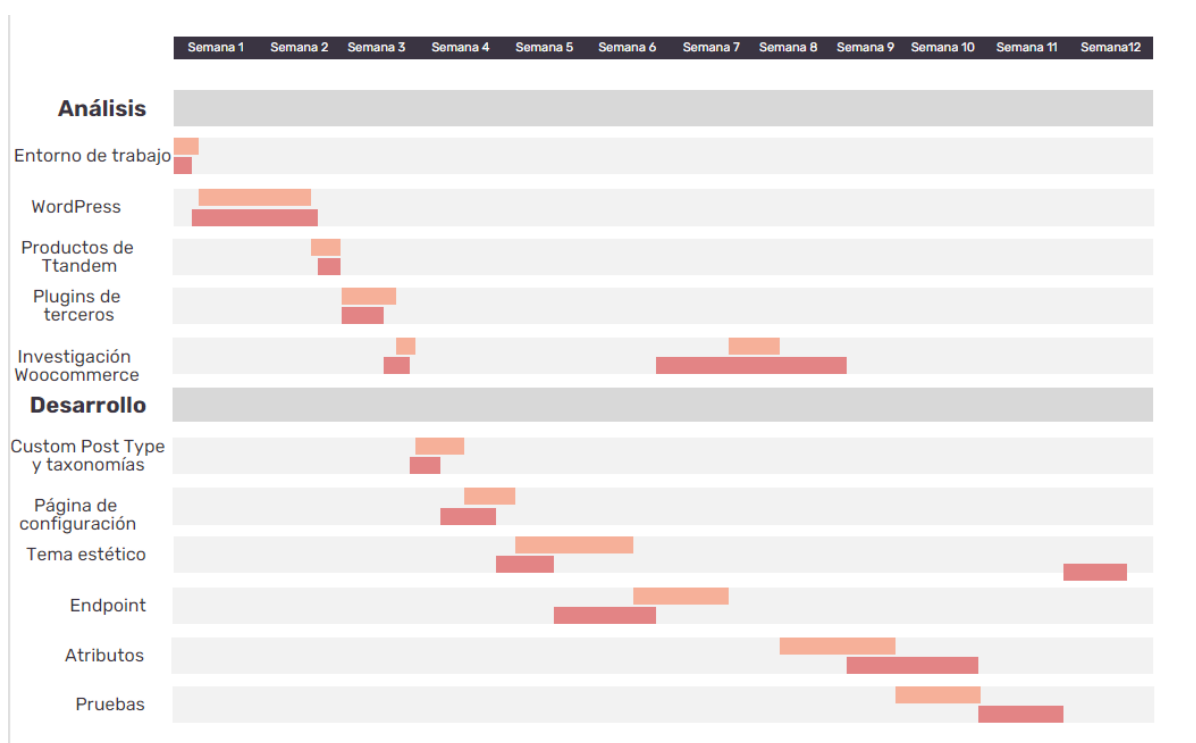


Diagrama 1. Diagrama de Gantt del proyecto.

Donde las barras de color naranja son el tiempo estimado por TTANDEM y el color rojo el tiempo real necesitado para completar cada apartado.

Como se puede apreciar en el diagrama de Gantt, el tiempo estimado para dos de los apartados fue inexacto por diversas causas que se exponen a continuación:

- ❖ Investigación de Woocommerce: Como se analiza de forma detallada en el apartado referente a la investigación de Woocommerce durante la fase de “Análisis y diseño”, se realizan dos periodos de investigación de este plugin durante el proyecto. La estimación de tiempo del primer periodo es aproximada en cuanto al que realmente se ha necesitado. En cuanto al segundo periodo, se esperaba una investigación relativamente corta para comprobar si era posible la implementación de atributos o no.

La razón principal del gran retraso es a causa de que, tras un primer estudio, se iba a llegar a la conclusión de que no era posible una implementación compatible de los atributos de Woocommerce. Pero durante el que se esperaba que fuera el último día dedicado a la investigación de este apartado, se encontró la primera información útil acerca de cómo podrían llegar a implementarse. A partir de este punto, el plan de proyecto se separó completamente del planteamiento inicial de tiempos estimados y se enfocó en mejorar la funcionalidad final del plugin en vez de cumplir los tiempos previstos.

- ❖ Tema estético: El tema estético se plantea en un principio como una tarea más del proyecto, la cual se desarrolla hasta completarla totalmente. Pero en realidad, como el tema no está ligado directamente con el plugin, se delega su creación a un segundo plano mientras se prioriza la funcionalidad del plugin. Por lo que, una vez creada la estructura general del tema, se aplaza su terminación y se prosigue con otras tareas.

Análisis y diseño

Al tener que crear un producto nuevo y en una plataforma desconocida se establece un periodo de investigación y aprendizaje. Por un lado, se debe obtener conocimiento suficiente sobre el desarrollo y funcionamiento de la plataforma WordPress, y por el otro, es necesario investigar hasta qué punto es realizable el catálogo propuesto.

Se propone conseguir las siguientes funcionalidades para el plugin:

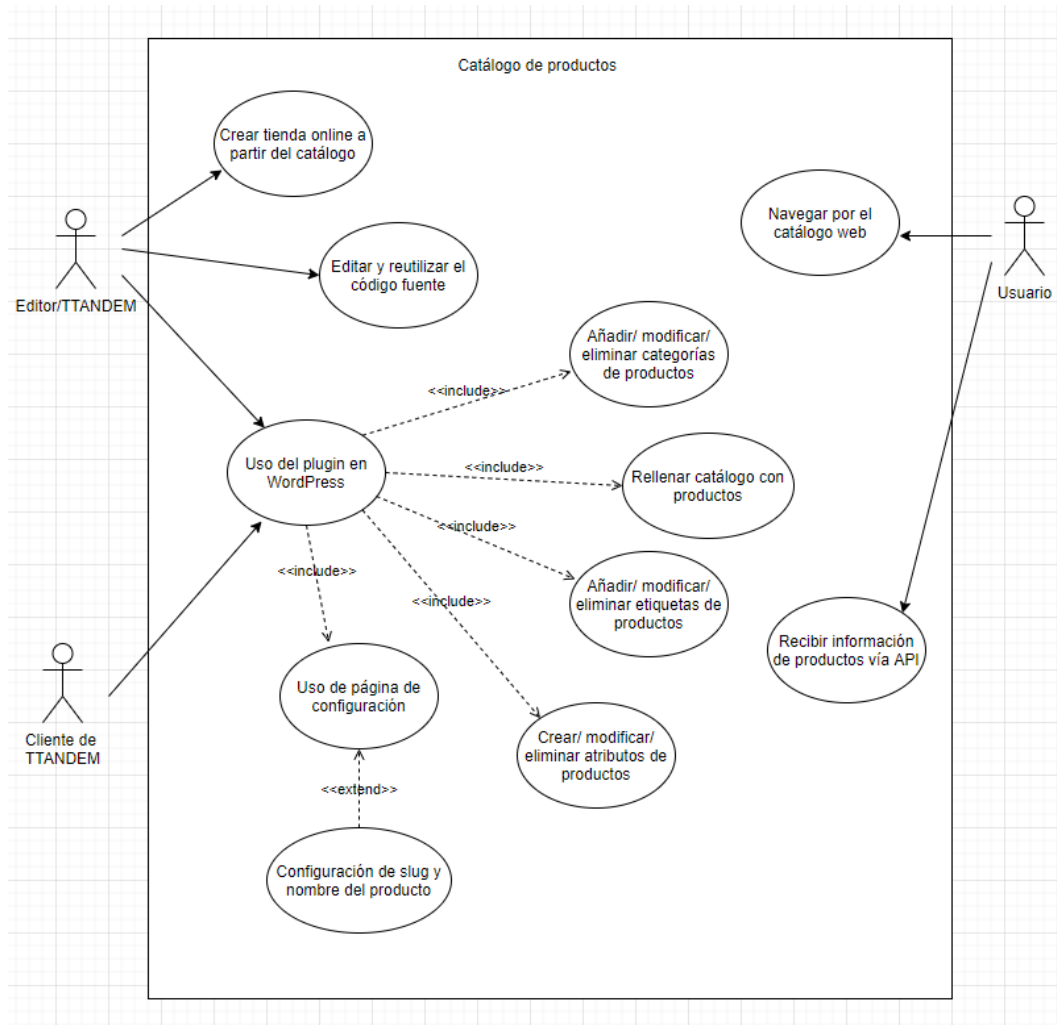


Diagrama 2. Diagrama de casos de uso del catálogo.

Configuración del entorno de trabajo

Como el producto se va a desarrollar en un entorno de trabajo personal se debe configurar dicho lugar con las herramientas adecuadas. Desde la empresa TTANDEM se proporciona la documentación necesaria para ello. Al ser WordPress una herramienta web es necesario instalar un programa de hosting virtual como Wamp o Xamp para no tener que contratar un servicio de hosting en el que desplegar constantemente el proyecto. Además, es indispensable instalar y configurar WordPress mismo. A parte de descargar los plugins necesarios para el desarrollo e investigación.

WordPress

WordPress es un sistema de gestión de contenidos o CMS. En un principio fue diseñado para crear y mantener blogs, pero hoy en día se usa para crear cualquier tipo de web. Principalmente sus contenidos se programan en PHP y HTML pero al ser un CMS tiene su gestión interna propia.

Antes de iniciar el desarrollo del producto se debía obtener el conocimiento necesario para poder trabajar en WordPress por lo que la primera semana de las prácticas se dedica completamente a su estudio. WordPress como tal tiene un codex (manual) online muy completo de donde es muy cómodo estudiar y entender todo lo necesario. No es posible comprender todas sus funcionalidades desde cero solo con lectura e investigación, por lo que se hace especial énfasis en las funciones necesarias para la creación del proyecto. Así que una vez habiendo interiorizado suficiente sobre su estructura básica, empiezan las primeras etapas de desarrollo y al mismo tiempo se van generando dudas nuevas.

Como resultado del estudio de WordPress se obtiene una idea básica de su funcionamiento, se enfoca el estudio en los plugins y temas ya que son los apartados con los que principalmente trabajan los desarrolladores. A continuación, se explican los apartados principales que son relevantes para la creación del producto final.

Plugins

Un plugin es un fragmento de código que aumenta la funcionalidad de WordPress, en otras palabras, una extensión. A partir de estos se pueden crear todo tipo de páginas web. Un plugin puede usarse para poder crear una tienda online como en el caso de Woocommerce o simplemente añadir una opción personalizada a la configuración de WordPress.

Para crear un plugin es necesario acceder a la carpeta “wp-content/plugins” dentro de WordPress y crear como mínimo un archivo con una función y un hook que llame a la función. Un plugin debe programarse en PHP, y puede incluir ficheros multimedia, CSS, HTML y Javascript... Adicionalmente, un plugin puede emplear una serie de funciones únicas ofrecidas por WordPress para trabajar con funcionalidades propias del CMS. Para obtener más información sobre estas funciones, se recomienda al lector acudir a la sección “Funciones de WordPress” de este mismo documento.

Los plugins pueden ser programados o descargados desde un repositorio ofrecido por WordPress de forma gratuita. Una vez descargados se pueden activar desde la pestaña “Plugins” en el menú de WordPress. WordPress cuenta con más de 48.000 plugins gratuitos, aunque también existen los plugins de pago de empresas privadas.



Figura 1. Plugin de Woocommerce desde la selección de WordPress.

Una vez activados, WordPress adquiere las funcionalidades del plugin.

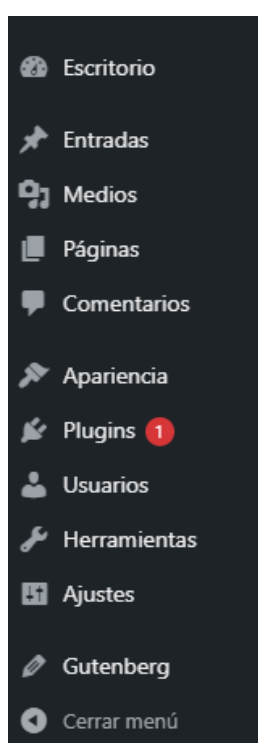


Figura 2. Menú de funcionalidades estándares de WordPress

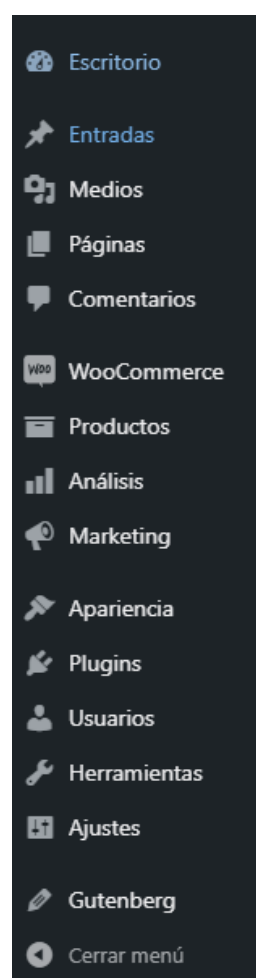


Figura 3. Mismo menú con el plugin Woocommerce activado

Como se puede ver en las figuras 2 y 3, al activar el plugin de Woocommerce se añaden los apartados de Woocommerce, Productos, Análisis y Marketing a WordPress.

Temas

También conocidos como “theme”, el tema es el código que se encarga de modificar cómo se muestran los contenidos de la página de WordPress. A diferencia del apartado estético de una página web fuera de WordPress, donde la estética se programa principalmente en CSS, en un tema de WordPress se usa tanto PHP para la arquitectura del sitio y archivos, como CSS para retoques meramente estéticos.

Se pueden emplear temas de dos maneras distintas, la primera de ellas supone un proceso similar al de la creación de un plugin, creando un tema completamente desde cero. En la carpeta “wp-content/themes”, se crea una carpeta con el nombre del tema y en su interior se crea un archivo php con el mismo nombre del tema. Añadiendo código HTML y CSS ya sería suficiente para que WordPress reconociese el tema. La segunda manera es creando un tema hijo y modificándolo. Un tema hijo es un tema que hereda las funcionalidades del tema padre del que se ha creado y permite modificarlas, mejorarlas y extenderlas. Para crearlo basta con copiar un tema ya existente en la carpeta “wp-content/themes”, por ejemplo “tema7”, pegar la copia en la misma carpeta y nombrarlo “tema7-child”. A continuación, dentro del tema hijo es necesario crear un archivo style.css, introducir la cabecera de temas que reconoce WordPress (véase figura 4) y rellenarlo con los datos que se quiera. Gracias a los nuevos datos con los que se rellena la cabecera, WordPress diferencia el tema original del nuevo tema. Una vez conseguido esto, se pueden modificar todos los elementos existentes en el tema hijo para configurarlo como se quiera.

```
1  /*!  
2  Theme Name:  
3  Theme URI:  
4  Author:  
5  Description:  
6  Version:  
7  Tested up to:  
8  Requires PHP:  
9  License:  
10 License URI:  
11 Tags:  
12 */
```

Figura 4. Cabecera para temas.

Los temas se pueden activar de manera similar a los plugins desde la pestaña “Apariencia” en el menú de WordPress.

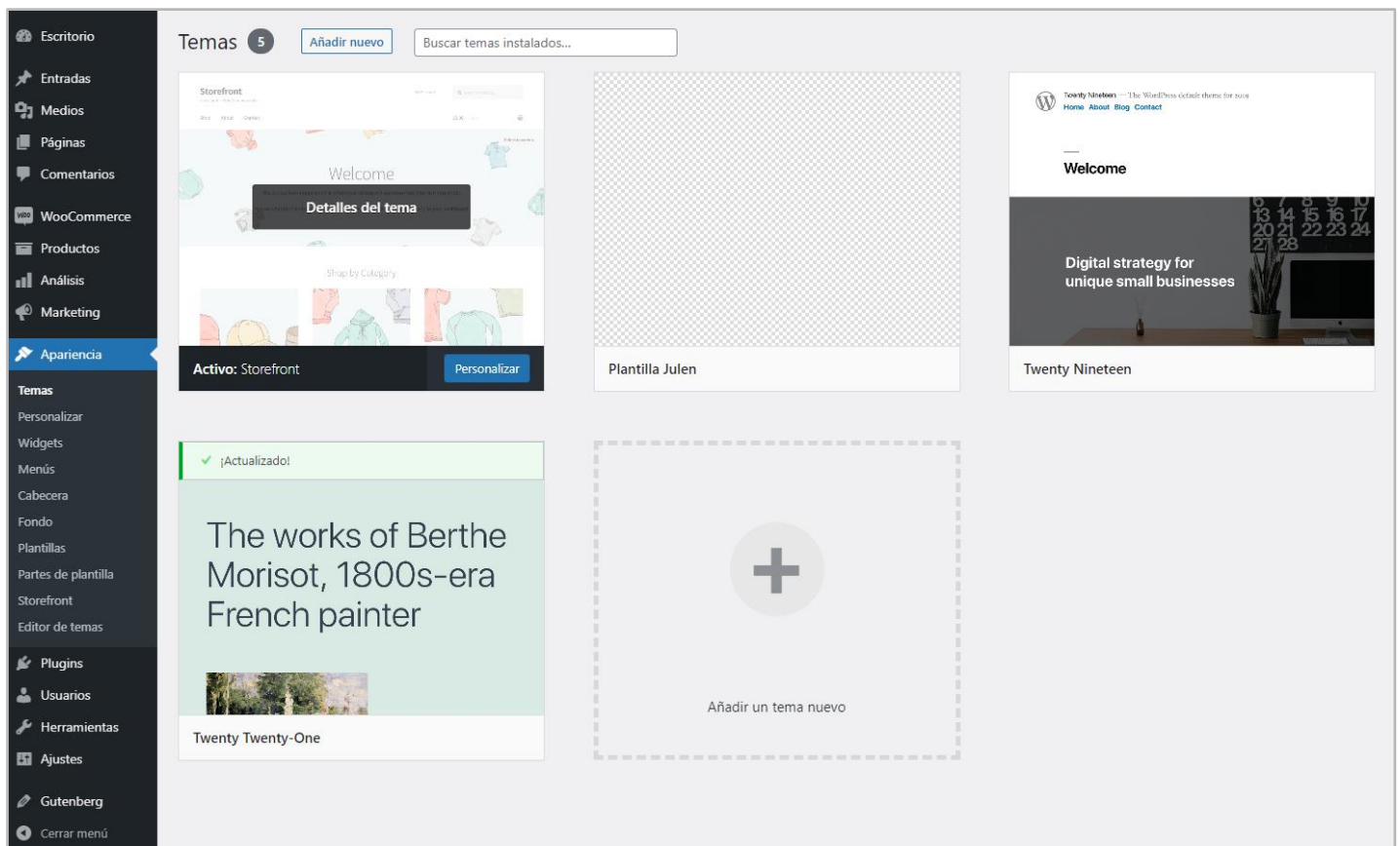


Figura 5. Ventana de selección de temas en WordPress

Los temas no están ligados a los plugins, un mismo tema puede usarse para mostrar diferentes contenidos y un plugin puede ser mostrado por distintos temas. Pero puede suceder que un tema no esté programado para mostrar todas las funcionalidades que un plugin añade. A continuación, se mostrará el contenido de WordPress con el plugin de Woocommerce activado con dos temas distintos para ver la diferencia de la programación de dos temas.

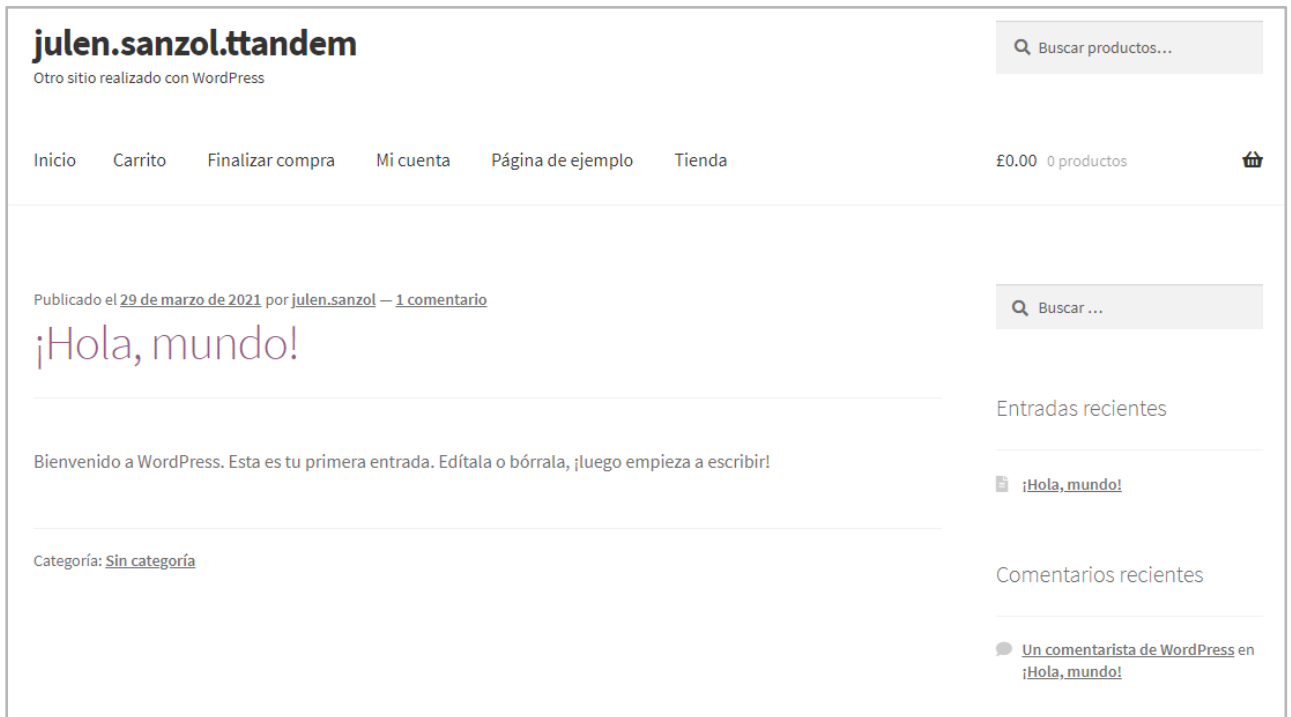


Figura 6. Tema de Woocommerce activado

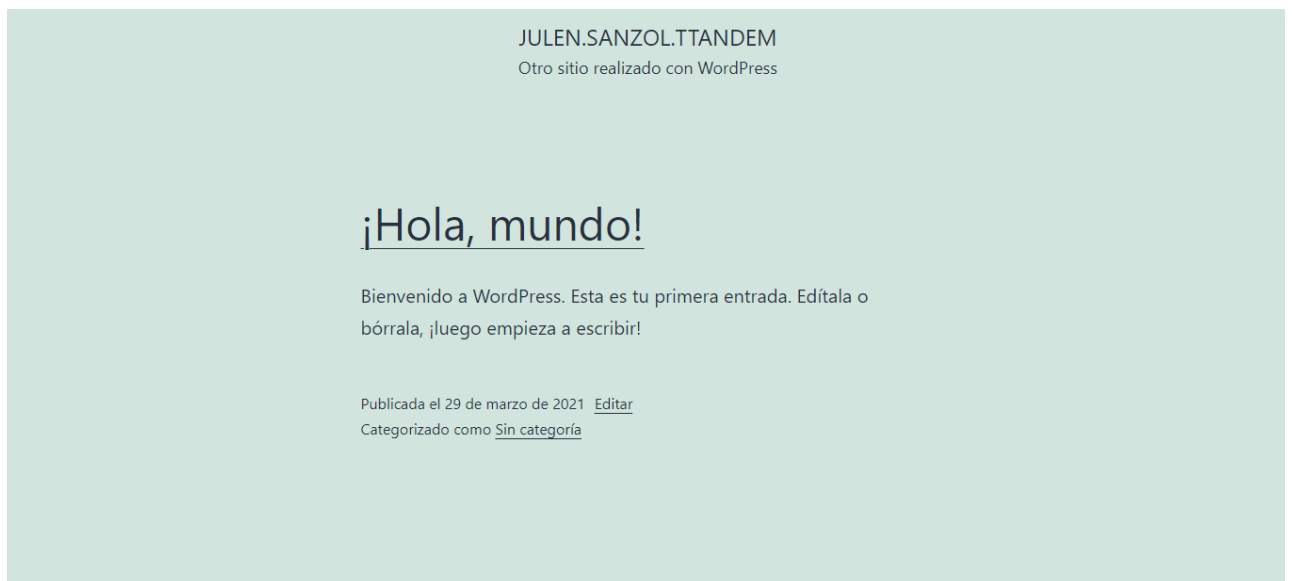


Figura 7. Tema Twenty Twenty-One de WordPress activado

Como se puede observar, se puede mostrar la página y sus contenidos con ambos temas activados. Pero el tema de Woocommerce está preparado para mostrar funcionalidades del plugin de Woocommerce, que están enfocadas al funcionamiento de una tienda online. El tema de WordPress por otro lado también muestra el post que viene por defecto en WordPress, pero no puede mostrar las funcionalidades del plugin activado ya que no está preparado para ello.

Funciones de WordPress

El funcionamiento de una página web de WordPress es diferente al de una página web convencional o una diseñada con la metodología Modelo-vista-controlador (MVC). WordPress añade las funcionalidades de los plugins mediante “ganchos” o “hooks” que se activan durante la ejecución de un evento del sistema. Un evento de WordPress hace referencia a un momento específico de la ejecución de WordPress, ejemplos de distintos eventos serían: Activación de un plugin, carga de la base de datos de WordPress, inicialización de la API de WordPress...

Hooks

Un hook es una acción interna de WordPress que sirve para añadir o modificar datos o su flujo durante la ejecución del sistema. Un hook funciona de la siguiente manera:

Se crea una función cualquiera y a continuación, mediante el uso de una función `action`, se crea el hook en el sistema. Posteriormente el sistema llama a dicha acción cuando el evento indicado en la función se activa.

```
function wporg_callback() {  
    // do something  
}  
add_action( 'init', 'wporg_callback' );
```

Figura 8. Ejemplo de creación de hook. [Codex de WordPress](#)[1].

La función `add_action` indica a WordPress que en el evento ‘init’ debe ejecutar la función `wporg_callback`. WordPress permite definir hooks ante multitud de eventos, en este caso ‘init’ es el momento después de que WordPress termine de cargarse, pero antes de empezar a cargar las cabeceras del tema.

`do_action(‘wporg_callback’)` ejecutaría la función directamente sin depender del estado en el que se encuentre el sistema, no depende de ningún evento y se ejecuta cuando se lee la función.

Y `add_filter` y `do_filter` tienen los mismos parámetros que las funciones `action` correspondientes, pero en vez de añadir una funcionalidad, los filter se usan para modificarlas.

En el producto de TTANDEM usa la función `add_action` para llamar a las funciones del plugin del catálogo y `add_filter` para modificar el funcionamiento de taxonomías de WordPress para que interactúen con los atributos que se explicarán más adelante.

Custom Post Type

Por defecto WordPress permite crear post para blogs, pero mediante plugins se pueden modificar para formar objetos al gusto. A estos post personalizados se les llama Custom Post Type.

Para la creación de Custom Post Types, existe en WordPress la función `register_post_type` al que se le pasa el nombre del nuevo objeto y un array de argumentos definidos previamente. como se muestra en este ejemplo de creación de un Custom Post Type de un libro:

```
function bf_register_custom_post_type() {
    /* Añado las etiquetas que aparecerán en el escritorio de WordPress */
    $labels = array(
        'name'           => _x( 'Libros', 'post type general name', 'text-domain' ),
        'singular_name'  => _x( 'Libro', 'post type singular name', 'text-domain' ),
        'menu_name'      => _x( 'Libros', 'admin menu', 'text-domain' ),
        'add_new'        => _x( 'Añadir nuevo', 'libro', 'text-domain' ),
        'add_new_item'   => __( 'Añadir nuevo libro', 'text-domain' ),
        'new_item'       => __( 'Nuevo libro', 'text-domain' ),
        'edit_item'      => __( 'Editar libro', 'text-domain' ),
        'view_item'      => __( 'Ver libro', 'text-domain' ),
        'all_items'      => __( 'Todos los libros', 'text-domain' ),
        'search_items'   => __( 'Buscar libros', 'text-domain' ),
        'not_found'      => __( 'No hay libros.', 'text-domain' ),
        'not_found_in_trash' => __( 'No hay libros en la papelera.', 'text-domain' )
    );

    /* Configuro el comportamiento y funcionalidades del nuevo custom post type */
    $args = array(
        'labels'          => $labels,
        'description'     => __( 'Descripción.', 'text-domain' ),
        'public'          => true,
        'publicly_queryable' => true,
        'show_ui'         => true,
        'show_in_menu'    => true,
        'query_var'       => true,
        'rewrite'         => array( 'slug' => 'libro' ),
        'capability_type' => 'post',
        'has_archive'     => true,
        'hierarchical'    => false,
        'menu_position'   => null,
        'supports'        => array( 'title', 'editor', 'author', 'thumbnail', 'excerpt',
    );

    register_post_type( 'libro', $args );
}
```

Figura 9. Ejemplo de creación de Custom Post Type. [Darío BF](#) [2].

Usualmente se proporcionan inputs para los argumentos que más importancia tienen y se dejan la mayoría con su valor por defecto, ya que existen más de 50 campos personalizables. Esta función se usará para crear los productos del catálogo.

Taxonomía

Se trata de un campo que sirve para agrupar Custom Post Types dentro de WordPress. Una taxonomía es creada con relación a un post y los agrupa en base a una característica concreta. Siguiendo con el ejemplo del Custom Post Type de un libro se creará una taxonomía para catalogarlos por su género:

```
function create_book_taxonomies() {  
    /* Configuramos las etiquetas que mostraremos en el escritorio de WordPress */  
    $labels = array(  
        'name'           => __( 'Géneros', 'taxonomy general name' ),  
        'singular_name'  => __( 'Género', 'taxonomy singular name' ),  
        'search_items'   => __( 'Buscar por Género' ),  
        'all_items'      => __( 'Todos los Géneros' ),  
        'parent_item'    => __( 'Género padre' ),  
        'parent_item_colon' => __( 'Género padre:' ),  
        'edit_item'       => __( 'Editar Género' ),  
        'update_item'    => __( 'Actualizar Género' ),  
        'add_new_item'   => __( 'Añadir nuevo Género' ),  
        'new_item_name'  => __( 'Nombre del nuevo Género' ),  
    );  
  
    /* Registramos la taxonomía y la configuramos como jerárquica (al estilo de las categorías) */  
    register_taxonomy( 'genero', array( 'libro' ), array(  
        'hierarchical'    => true,  
        'labels'           => $labels,  
        'show_ui'         => true,  
        'query_var'       => true,  
        'rewrite'         => array( 'slug' => 'genero' ),  
    ));  
}
```

Figura 10. Ejemplo de creación de taxonomía. [Darío BF](#) [2].

Como se muestra, la función de WordPress `register_taxonomy` crea la nueva taxonomía en relación con el Custom Post Type libro creado anteriormente, y permite agrupar los libros por género. Cuando se crea una taxonomía, dentro de WordPress es posible crear los grupos de búsqueda dentro de esta, como el grupo “comedia” para este ejemplo concreto, y posteriormente en un objeto libro se puede seleccionar dicho grupo para relacionar ese libro concreto con la taxonomía género y el grupo comedia. En el catálogo se crearán taxonomías para agrupar los productos. Por ejemplo, en base a su categoría como ropa, calzado, etc.

Slug

La palabra slug hace referencia al texto que se emplea en la URL del sitio web después del nombre del dominio de WordPress. Cuando se crea un Custom Post Type, es recomendable usar un slug concreto para poder acceder mediante la URL al tipo de post. En los ejemplos anteriores al crear el Custom Post Type libro, se ordena a WordPress que sobrescriba (con el argumento ‘rewrite’) el slug del objeto por ‘libro’, y lo mismo con la taxonomía género.

En el catálogo de productos se usará como ejemplo el slug “product” para el Custom Post Type de los productos, por lo tanto, para acceder a un producto llamado “zapatilla”, la URL del productos sería:

`https://dominio/wordpress/product/zapatilla`

Base de datos de WordPress

WordPress almacena sus datos en una base de datos MySQL y esta es administrada mediante phpMyAdmin. Dicha base de datos se despliega de forma automática cuando se instala WordPress y almacena todos los datos que el sistema requiere para operar.

Los plugins también pueden manipular o usar la base de datos a su antojo, aunque modificar las tablas propias de WordPress no es recomendable, ya que puede crear graves problemas de funcionamiento. Cuando un plugin crea una tabla en la base de datos, al desactivar el plugin dicha tabla no se elimina.

Para la comunicación con la base de datos, WordPress usa una variable global `$wpdb` y funciones predefinidas. `Wpdb` es la clase que contiene un conjunto de funciones y variables preparadas para interactuar con la base de datos. Esto facilita mucho el desarrollo de funciones que interactúan con la base de datos en el plugin.

API de WordPress

WordPress dispone de una REST API que permite la comunicación directa con su base de datos desde diferentes programas. Esta comunicación se produce mediante URLs también conocidas como rutas de acceso o endpoints. Por defecto, WordPress tiene unos endpoints definidos, pero también permite crear endpoints customizados que trabajen con los datos de los plugins creados.

```
add_action( 'rest_api_init', function () {
    register_rest_route( 'myplugin/v1', '/author/(?P<id>\d+)', array(
        'methods' => 'GET',
        'callback' => 'my_awesome_func',
        'args' => array(
            'id' => array(
                'validate_callback' => 'is_numeric'
            ),
        ),
        'permission_callback' => function () {
            return current_user_can( 'edit_others_posts' );
        }
    ) );
} );
```

Figura 11. Ejemplo creación de un nuevo endpoint en wordpress. [Codex de WordPress](#)

Como se puede ver en la figura 11, la primera línea crea el hook que se activa cuando la api de WordPress termina de cargarse. A continuación, registra la nueva ruta o endpoint en la api, define con ‘callback’ lo que hará dicho endpoint y finalmente asigna los permisos de acceso con ‘permission_callback’.

Estudio de productos anteriores de TTANDEM

Con la intención de identificar y definir los requisitos que debe cumplir la plantilla del catálogo, TTANDEM proporciona varios proyectos de catálogos desarrollados anteriormente para clientes reales. Durante esta fase de estudio se recorren estas implementaciones en busca de patrones comunes y prácticas habituales dentro de la empresa.

Como resultado de la investigación se define que ciertos componentes de la web dependerán de los productos y servicios que quiera ofrecer la empresa del cliente. Al igual que el encabezado, pie y estética general de la página dependerán también del cliente a no ser que se especifique lo contrario en su contrato.

El catálogo, que sería la parte fundamental del proyecto, debe ser claro y de uso intuitivo. El hecho de filtrar los productos en la página principal o separarlos por páginas de tipo es secundario y simplemente estético. Además, un producto en el catálogo debe diferenciarse claramente con la imagen y descripción proporcionadas. En cuanto a la página del producto debe ser detallada y muy completa.

Por lo tanto, se define como requisito mínimo para crear la plantilla del catálogo, que esta debe disponer de productos. Como requisitos secundarios se propone la creación de agrupaciones para búsquedas de dichos productos. El tema estético debe ser severamente simplificado, ya que la estética final dependerá enteramente de los distintos posibles clientes finales y sus gustos y preferencias.

Se adjunta a la presente memoria, como “ANEXO 2”, el documento generado durante el análisis de los proyectos de TTANDEM.

Plugins de terceros

Durante el desarrollo del producto se han utilizado dos plugins de WordPress distintos para mejorar la experiencia del programador. Haciendo uso de documentación proporcionada por TTANDEM y búsquedas propias en internet se estudia cada uno de los plugins durante las primeras semanas intercalando con pruebas de uso para acostumbrarse a la programación en WordPress.

Gutenberg

Es un editor de texto preparado para WordPress que permite gestionar y construir contenidos mediante bloques de forma muy sencilla. Permite modificar la forma original de mostrar el contenido que tiene WordPress.

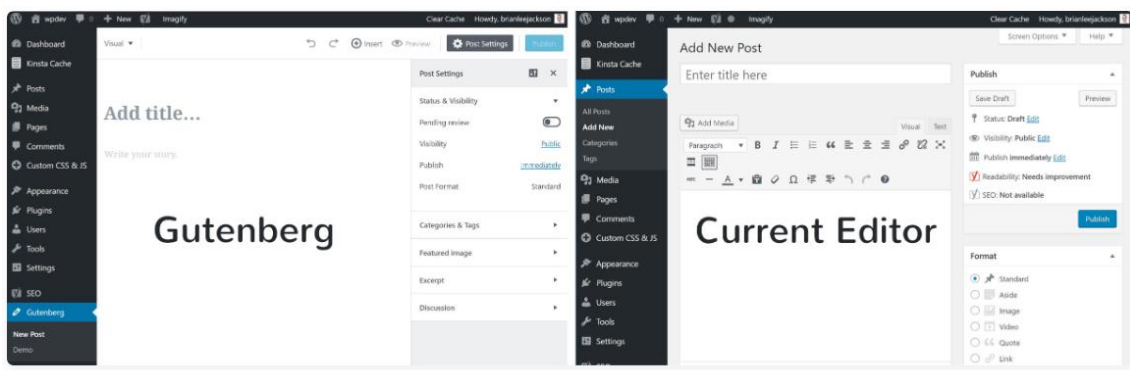


Figura 12. Comparación de Gutenberg vs editor actual. [Blog de Brian Jackson](#) [3].

CMB2

CMB2 es otro plugin de WordPress que implementa un conjunto de herramientas para personalizar y modificar campos de los Custom Post Types y taxonomías. Añade metabox, campos personalizados y formularios que dan a los contenidos del plugin la opción de ser modificados sin necesidad de tener que tocar el código.

Se ha investigado este plugin con la finalidad de añadir funcionalidad que permitan modificar aspectos básicos del catálogo. Como resultado de la investigación y teniendo en cuenta los requisitos de la plantilla del catálogo se decidió que la forma más eficiente de implementar dichos campos es mediante la creación de una página de opciones. Esta sección de configuración supondrá un punto único de personalización mediante el que habilitar distintos metabox para permitir modificar el slug y el nombre público del producto. En caso de implementar más funcionalidades en el futuro se añadirían los metabox para modificar los slugs nuevos.

Estudio de compatibilidad de Woocommerce

Woocommerce es un módulo de eCommerce que se integra en WordPress como plugin. ECommerce o comercio electrónico es el modelo de negocio de tienda virtual, no solamente mediante páginas web, sino también a través de las redes sociales.

Se trata de uno de los plugins más populares de WordPress, es tal su popularidad que en el año 2015 la empresa encargada de WordPress compró el plugin por 30 millones de dólares.

Su función principal es la de permitir crear tiendas online, funciona como una plantilla para ello, pero las tiendas resultantes suelen ser muy parecidas entre sí, a no ser que haya un programador web por medio.

La investigación de los requisitos de compatibilidad con Woocommerce se separan en dos periodos:

El primero, al comienzo del proyecto, se descubren y evalúan las funcionalidades que se pueden programar en el catálogo manteniendo la compatibilidad con un producto de Woocommerce, y en segundo lugar, se investigan funcionalidades más complejas que durante la primera fase no se pudieron entender e implementar.

Una vez finalizadas todas las fases de investigación, se desea crear el siguiente ejemplo de producto para el catálogo.



Zapatilla azul

Es una zapatilla deportiva de color azul.

Categorías

- ☒ Ropa
- ☒ Calzado
- ☒ Calzado deportivo
- ☐ Sandalias
- ☐ Sudaderas
- ☐ Comida

Etiquetas

- ☒ Deporte
- ☒ Verano
- ☐ Montaña

Atributos del producto

Color del producto

- ☒ Azul
- ☐ Rojo
- ☐ Verde

Talla Calzado del producto

- ☒ 37
- ☒ 39
- ☒ 40
- ☐ 38

Material del producto

- ☐ Algodón
- ☒ Seda

Figura 13. Ejemplo de objeto que se desea crear para el catálogo.

El producto está compuesto por un título, una foto o varias del producto y una descripción. En cuanto a índices de agrupamientos se configuran las categorías del producto y las etiquetas como se muestra, añadiendo los subgrupos que se desee. Y los atributos son ejemplos, ya que pueden crearse todos los que se quiera. Cabe remarcar que los grupos de búsqueda pueden ser estructurados por jerarquías como se muestra la categoría ropa de la imagen. Woocommerce presenta una estructura similar para sus productos.

Primer periodo de investigación

Antes de empezar a desarrollar el producto final se investiga la estructura que Woocommerce les da a sus productos y cómo los almacena. Tras investigar el código del plugin de e-commerce y su funcionamiento, se llega a los siguientes resultados:

Para crear un producto, Woocommerce crea un Custom Post Type mediante la función `register_post_type` explicada anteriormente. En dicha función asigna como nombre una palabra clave al nuevo tipo de post, lo cual hace que todo post de este nuevo tipo creado sea reconocido como un producto de la tienda.

Como prueba, en nuestro catálogo, se usa la misma palabra clave al crear el Custom Post Type del producto, pero se asigna una configuración diferente de argumentos. Dicha configuración contiene menos argumentos, ya que es necesario mantener el

requisito de crear la plantilla de un catálogo, por lo que se evita que el producto esté configurado en exceso. Posteriormente, al cambiar del catálogo de TTANDEM a Woocommerce, se confirma que automáticamente reconoce los productos creados con el nuevo Custom Post Type y permite trabajar con ellos. Confirmamos de este modo que es posible crear un producto compatible con Woocommerce, lo cual permite cumplir el requisito mínimo para crear un catálogo de productos, programar los productos.

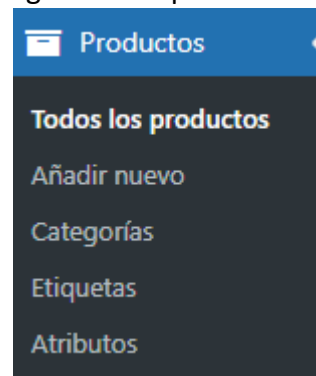


Figura 14. Submenú de los productos en Woocommerce

Tras esa primera compatibilidad descubierta se plantea investigar la parte relacionada con las taxonomías de productos. Se descubre la taxonomía de categorías y la de etiquetas de productos siguen una configuración similar a la del Custom Post Type del producto: en la función de WordPress de creación de una nueva taxonomía se emplea una palabra clave para cada una de las taxonomías que Woocommerce reconoce de forma automática. Lo que permite implementar dos de las tres taxonomías usadas en Woocommerce, dejando la taxonomía atributos para una investigación posterior.

Estas dos taxonomías son las encargadas de agrupar los productos por categorías y por etiquetas. Las categorías de los productos son aquellas que separan varios productos por tipos, ya sea ropa por ejemplo y dentro de la ropa el calzado, camisetas, etc... Las etiquetas en cambio son agrupaciones de productos enfocados a un mismo ámbito, como por ejemplo productos para la playa, de verano, etc. Ya que la implementación de ambas es similar a la del producto no se ve necesario hacer pruebas de comprobación y se deja directamente para la fase de desarrollo.

Segundo periodo de investigación

La segunda parte de la investigación de Woocommerce comienza una vez finalizada la implementación tanto de los productos, como de las taxonomías de categorías y etiquetas. Ahora es necesario indagar la forma de programar la taxonomía de Woocommerce que no se pudo conseguir durante el primer periodo de investigación, los atributos de los productos, ya que estos no tienen una estructuración simple como las anteriores. Los atributos de un producto son las características que diferencian un producto de otro similar. Ejemplos de atributos serían los colores o tallas.

A diferencia de las otras dos taxonomías (categorías y etiquetas), donde se crean con una palabra clave y Woocommerce las reconoce por ello, en las taxonomías de atributos cada atributo es una taxonomía única. Es decir, en Woocommerce se

pueden crear tantos atributos que se deseen, y por cada atributo creado en el plugin, en WordPress genera una taxonomía. Por lo tanto, Woocommerce no puede reconocer las taxonomías creadas para los atributos usando palabras clave porque el usuario puede crear infinitos atributos. Lo que confirma que usa otro sistema distinto al de las categorías y etiquetas para reconocer las taxonomías de atributos.

La segunda gran diferencia que tienen los atributos es que, como cada atributo es una taxonomía en sí, Woocommerce no los reconoce por una palabra clave como pasaba con el resto de los campos. Por ello hay que investigar cómo hace Woocommerce para crear y almacenar estas taxonomías de atributos, y una vez descubierto esto, cómo hace para reconocer las taxonomías de los atributos, si no es por palabras clave.

Resultado de investigación de atributos

Se investiga todo el código de Woocommerce, junto con su comportamiento cuando se usan los atributos en el plugin y también la documentación disponible en la página web de Woocommerce para lograr idear una posible forma de programarlos.

Se descubre que Woocommerce crea diversas tablas en la base de datos de WordPress, y que una de esas tablas guarda exclusivamente los nombres de los atributos creados con el plugin de Woocommerce. Solamente el nombre, no guarda ningún grupo de búsqueda ni ninguna información adicional de ningún tipo. Gracias a esto se sigue la pista en el código a las llamadas a dicha tabla de la base de datos y se logra encontrar finalmente el funcionamiento que Woocommerce tiene respecto a los atributos.

Al crear un nuevo atributo desde la interfaz de usuario de WordPress, Woocommerce agrega a la tabla de atributos el nombre del nuevo atributo. Una vez guardado, otra función se encarga de leer los nombres de esta misma tabla y crear una taxonomía con cada nombre almacenado. Por último, los grupos dentro de cada taxonomía de los atributos se almacenan en las tablas de WordPress en la base de datos como si de cualquier otra taxonomía se tratase.

Una vez comprendido el funcionamiento ya se puede empezar a desarrollar código que intentará comportarse de manera similar a como lo hace Woocommerce en el catálogo.

Desarrollo del proyecto

Tras la investigación previa, se crea un plan de proyecto inicial y se obtiene un desglose de desarrollo en cuatro fases. Esto permite implementar una primera versión del producto que cumpla los requisitos estudiados y que logre los hitos establecidos en los objetivos. Una vez completadas esas cuatro fases, se procederá a continuar la investigación que se referencia como segundo periodo en el apartado de “Análisis y diseño”. Esta segunda fase conlleva añadir una fase extra de desarrollo para ampliar el proyecto antes de comenzar con las pruebas.

Fase 1: Creación del Custom Post Type con dos taxonomías

Se propone implementar una estructura básica de los productos del catálogo junto con los índices de agrupamiento de categorías y etiquetas, conservando en todo momento la compatibilidad de todos los elementos con Woocommerce. Para esto se emplean los resultados obtenidos durante el primer periodo de investigación.

Primero se crea el plugin creando un fichero con el nombre del proyecto “TTANDEM” dentro del directorio de plugins de WordPress. A dicha carpeta se le añade un primer archivo php con el mismo nombre que el plugin y se programa un primer hook que llamará a la función que se encargará de la mayoría de los hooks del plugin. Esto facilitará la corrección de errores durante el desarrollo del plugin y tener casi todos los hooks en una misma función facilita a TTANDEM trabajar con el código para proyectos de futuros clientes. Las funciones involucradas en ambas fases de desarrollo se programan en distintos archivos y se llamarán mediante hooks desde la función de hooks descrita.

Por lo tanto, en un archivo php aparte, se programa la creación de un Custom Post Type para los productos del catálogo, similar al ejemplo de la figura 9, enfocando cada label y argumento de la función a un producto de catálogo. También se define un slug, “producto” por el momento ya que se especifica en los objetivos que habrá que implementar opciones de edición para este campo más adelante. Y por último se registra el Custom Post Type usando la misma palabra clave que usa Woocommerce.

Una vez hecho esto, se generan dos taxonomías, una para las categorías y otra para las etiquetas, relacionadas con el nuevo Custom Post Type creado. La estructura de ambas taxonomías es similar a la figura 10, se añaden las descripciones respectivas a categorías y etiquetas de productos y se escoge un slug temporal para cada una a la espera de implementar la página de configuración que permita su renombrado.

La finalización de esta fase llega sin ningún contratiempo y se crean productos piloto y taxonomías para futuras pruebas de transición entre plugins.

Fase 2: Página de configuración

Es necesario crear una página de configuración de slugs y del nombre general del producto. Como el objetivo final es crear una plantilla de un catálogo, el fin de esta página es posibilitar a los clientes e incluso a TTANDEM la modificación de los campos que es imposible no programar durante el desarrollo de este proyecto, sin necesidad de alterar el código fuente.

Una vez creada la estructura principal del plugin y el Custom Post Type del producto junto con las dos taxonomías, se crea una página de configuración básica para los slugs y el nombre principal del producto.

Usando el plugin instalado CMB2 se implementa una página de configuración, donde se crean 4 metabox que permiten recoger los nuevos slugs y el nuevo nombre del producto.

```
11 class TTPC_cmb2{
12
13     public static function product_create_option_page() {
14
15         $cmb_options = new_cmb2_box( array(
16             'id' => 'slug_editor',
17             'title' => __( 'slug_editor', 'product' ),
18             'object_types' => array( 'options-page' ), // tipo de estructura del objeto cmb2
19             'option_key' => 'slug_editor',
20             'icon_url' => 'dashicons-admin-generic', // Menu icon
21             'capability' => 'edit_posts', // Capability required to view this options page.
22             'position' => 6, // Menu position.
23             'save_button' => 'Save',
24         ) );
25
26         $cmb_options->add_field( array(
27             'name' => __( 'Editar slug de los productos', 'products' ),
28             'desc' => __( 'Modifica el valor del producto en la url', 'product' ),
29             'id' => 'edit_CPT_slug',
30             'type' => 'text',
31         ) );
32
33         $cmb_options->add_field( array(
34             'name' => __( 'Editar slug de la taxonomía categoría', 'products' ),
35             'desc' => __( 'Modifica el valor de la taxonomía de categoría del producto en la url', 'product' ),
36             'id' => 'edit_taxonomy_category_slug',
37             'type' => 'text',
38         ) );
39
40         $cmb_options->add_field( array(
41             'name' => __( 'Editar slug de la taxonomía Etiqueta', 'products' ),
42             'desc' => __( 'Modifica el valor de la taxonomía de la etiqueta del producto en la url', 'product' ),
43             'id' => 'edit_taxonomy_tag_slug',
44             'type' => 'text',
45         ) );
46
47         $cmb_options->add_field( array(
48             'name' => __( 'Editar el nombre del producto', 'products' ),
49             'desc' => __( 'Modifica el nombre que se muestra publicamente del producto', 'product' ),
50             'id' => 'edit_CPT_label',
51             'type' => 'text',
52         ) );
53     }
54 }
55 }
```

Figura 15. Código de la página de configuración.

Mediante la función “new_cmb2_box” de la línea 15 se crea el nuevo objeto de configuración de opciones y en la línea 19 se especifica que dicho objeto será una página de configuración. A continuación, se crean los cuatro metabox, uno para cada apartado que se desea poder modificar. El resultado final de la página en WordPress es el siguiente:

| | | |
|---------------------------------------|---|---|
| Editar slug de los productos | <input type="text" value="products"/> | Modifica el valor del producto en la url |
| Editar slug de la taxonomía categoría | <input type="text" value="categories"/> | Modifica el valor de la taxonomía de categoría del producto en la url |
| Editar slug de la taxonomía etiqueta | <input type="text" value="tags"/> | Modifica el valor de la taxonomía de la etiqueta del producto en la url |
| Editar el nombre del producto | <input type="text" value="Bicicleta"/> | Modifica el nombre que se muestra publicamente del producto |

Figura 16. Página de configuración.

Una vez implementada esta página de configuración, se debe modificar el código de creación de Custom Post Type y taxonomías para recibir los valores de la página. Primero se verifica que existen valores guardados en la variable que recoge los datos. Y posteriormente se reemplazan todos los slug del argumento “rewrite” por la variable que contiene el nuevo slug y lo mismo con el nombre público del producto.

```
'rewrite' => array('slug' => $slug_CPT),
```

Figura 17 .El slug se sobrescribe mediante el valor de la variable.

Fase 3: Tema estético o plantilla

Es necesario crear un tema propio en WordPress para facilitar la demostración de los objetos creados y dar forma a lo que podría ser el catálogo final.

Para la creación del tema se usa el método de tema hijo, y a partir de los archivos básicos que venían en el tema padre se generan los siguientes archivos php que componen la estructura de un tema de WordPress:

- 404: Se encarga de mostrar un mensaje de error cuando se produce el error 404.
- archive: Se ocupa de que en caso de que existan productos creados, llamar al fichero que se encarga de mostrar los productos en el menú, “content-archive”. En caso de no existir productos, muestra el contenido de “content-none”.
- content-archive: Estructura como se muestra la información de los productos en la página de búsqueda.
- content-none: Muestra el mensaje de que no existen posts.

- footer: Muestra y estructura el contenido del pie de página en todas las páginas, es posible configurar un footer distinto para diferentes apartados de la página web.
- functions: Contiene las funciones de configuración del tema como por ejemplo los hook que se encarga de añadir un sidebar o la función encargada de leer la versión del tema del style.css.
- header: Tiene la misma funcionalidad que el footer pero se encarga de los encabezados. En este caso añadimos un navigation bar usando la función “wp_nav_menu” y un hook de add_filter.
- home: Es la primera página que muestra el tema cuando se carga, se ha programado para que muestre los últimos 4 productos añadidos al catálogo mediante una “WP_query” que se ocupa de la búsqueda de los Custom Post Type del tipo producto que se ha creado en el plugin.
- sidebar: añade un sidebar al tema mediante la función de WordPress “dynamic_sidebar”.
- single: Es el archivo responsable de que cuando se intenta acceder a un producto, si existe, llamar al fichero “content-single” para que muestre el producto.
- content-single: Estructura como se muestra el contenido de un producto cuando se accede a él.
- editor-style-classic.css: Contiene todo el código css referente a la estética del tema.

No se añaden funcionalidades extra que mostrar o detalles al tema, ya que su funcionalidad en el proyecto es mostrar los productos que se añaden al catálogo.



Figura 18. Muestra de resultado del tema con un producto piloto.

Fase 4: REST API

Se desea implementar la posibilidad de acceder a los datos de los productos del catálogo a través de la api de WordPress, por lo que se debe crear un endpoint en la API que devuelva la información de los productos. Es posible programar un endpoint que solo lea los datos o que también pueda modificarlos, en este caso para cumplir los objetivos del proyecto es necesario crear un endpoint solo de lectura.

El endpoint está orientado a estudiar la posibilidad de enviar datos relacionados con los productos creados en el catálogo fuera de WordPress y que, al mismo tiempo, estos productos en todo momento sean compatibles con Woocommerce. A causa de esto la ruta del endpoint se define idéntica a la que Woocommerce usa para completar la misma función en la API de WordPress.

Se programa un endpoint de forma que un hook activado durante 'rest_api_init', fin de carga e inicio de la rest api, llame a una función que registra el nuevo endpoint o ruta de acceso. Además, en esta función se define el método que se usará para obtener los datos, GET en nuestro caso, los permisos de acceso y la función de callback.

```
'methods' => 'GET',  
'callback' => 'get_products_by_category',  
'permission_callback' => function () {  
    return current_user_can( 'edit_others_posts' );  
}
```

Figura 19. Parte del código de la función que añade el endpoint.

La función que se llama en callback especifica los datos que devuelve el endpoint. El parámetro `permission_callback` especifica que el que use la API necesitará identificarse

con el usuario de WordPress y se comprobará si el usuario tiene permisos de edición de post, lo que quiere decir que solo devolverá información si el usuario que se identifica es un editor.

La función de callback que se muestra en la figura 19 recibe un parámetro que define los productos de qué categoría tiene que devolver dato el API, y además en caso de no existir dicho parámetro, devuelve todos los productos existentes en el catálogo. Todos los datos devueltos en cualquiera de los casos son devueltos en formato JSON.

Fase 5: Atributos de Woocommerce

En una tienda de productos, lo más importante antes de comprar algo son sus características: color, talla, material... En Woocommerce dichas características se almacenan como atributos, donde cada atributo es una taxonomía. El objetivo es lograr crear unos atributos similares a los de Woocommerce en el catálogo de TTANDEM. Estos atributos deben ser reconocidos automáticamente por Woocommerce al cambiar del plugin de TTANDEM al suyo.

En base a los descubrimientos expuestos en el segundo periodo de investigación de la fase de “Análisis y diseño” se comienza el desarrollo de los atributos. Durante el desarrollo, es necesario activar el plugin Woocommerce para crear por primera vez la tabla de atributos en la base de datos y rellenarla, porque por el momento no se dispone de nada desarrollado respecto a los atributos en nuestro plugin.

1. Se activa el plugin de Woocommerce en Wordpress y se crean varios atributos. Esto ayudará a tener ciertos datos creados en la tabla de la base de datos de atributos de Woocommerce para comprobar durante el desarrollo si las funciones que se van programando leen correctamente los datos de esa tabla.
2. Se desactiva el plugin de Woocommerce ya que no lo necesitaremos más y se activa el plugin de catálogo de TTANDEM. A continuación, se programa una función que permita leer los datos almacenados en la tabla de atributos.
3. Basándose en las funciones de Woocommerce que trabajan con atributos, se añade un nuevo método a nuestro proyecto. Esta función recoge los nombres de atributos de la tabla que recibe la función del paso 2, y crea una taxonomía de WordPress mediante “register_taxonomy()” con cada nombre. Con la diferencia de que se reemplazan los argumentos por defecto de WordPress por otros específicos creados para TTANDEM.

Como resultado, los atributos son convertidos a taxonomías y se les puede añadir subgrupos de búsqueda como si de una taxonomía por defecto de WordPress se tratara. Pero su estructura es algo diferente, lo que permite que al activar Woocommerce de nuevo, este reconozca las taxonomías de los atributos y también los subgrupos de búsqueda que se crean desde WordPress en cada taxonomía.

4. Se añade al plugin de TTANDEM la misma página de creación de atributos que usa Woocommerce. Esta página es una interfaz que permite crear nuevo atributo, y al crearlo almacena los nuevos datos en la tabla de atributos de la base de datos. También posibilita acceder a las taxonomías de los atributos ya creados para agregar grupos de búsqueda.

Como resultado, el plugin de catálogos ya puede leer y almacenar datos en la tabla de atributos de Woocommerce sin necesidad de tener activado su plugin. Y también es capaz de crear taxonomías en base a los atributos existentes en la base de datos.

En las siguientes imágenes, se muestran las pantallas de creación de nuevos atributos, el acceso a los atributos ya existentes y la creación de subgrupos dentro de cada atributo.

Así pues, se da por finalizada la fase de desarrollo de los atributos como exitosa. En la sección de pruebas del presente documento se exponen las casuísticas en las que podría alcanzarse un *estado de error*.

Figura 20. Página de creación de atributos en el catálogo.

| Nombre | Slug | Ordenar por | Términos |
|-----------------------|---------------|---------------------|--|
| Color | color | Orden personalizado | Rojo, Verde Configurar términos |
| Talla pantalón | tallapantalon | Orden personalizado | 33, 43, 44 Configurar términos |

Figura 21. Apartado de acceso a atributos existentes en la página de atributos.

Color del producto

Añadir nuevo Color

Nombre

El nombre es cómo aparecerá en tu sitio.

Slug

El «slug» es la versión amigable de la URL para el nombre. Suele estar en minúsculas y contiene solo letras, números y guiones.

Padre Color

Ninguna ▾

Asigna un término superior para crear una jerarquía. El término jazz, por ejemplo, sería el superior de bebop y big band.

Descripción

La descripción no suele mostrarse por defecto, sin embargo hay algunos temas que puede que la muestren.

Añadir nuevo Color

Acciones en lote ▾ **Aplicar**

| <input type="checkbox"/> | Nombre | Descripción | Slug |
|--------------------------|--------|------------------------|-------|
| <input type="checkbox"/> | Rojo | Producto de color rojo | rojo |
| <input type="checkbox"/> | Verde | Es color verde | verde |

Acciones en lote ▾ **Aplicar**

Figura 22. Página de configuración de un atributo.

Fase de pruebas y resultados

Al finalizar cada fase de desarrollo, se realizan comprobaciones rutinarias para verificar que no se han cometido errores graves. Estas pruebas han consistido en comprobar que WordPress reconoce las funcionalidades añadidas en cada fase y que el uso de dichas funcionalidades no provocasen errores de ejecución.

En cambio, una vez desarrolladas todas las funcionalidades del producto final, se inicia una fase de pruebas concisa para comprobar que el funcionamiento de todo apartado es correcto y cumple con los requisitos establecidos en los objetivos. Primero se realizan comprobaciones de funcionamiento de apartados generales, y tras cada resultado exitoso se van sometiendo partes cada vez más específicas a test.

Pruebas del producto y taxonomías de categorías y etiquetas

1. Se comprueba mediante la interfaz de WordPress que las tres funcionalidades han sido añadidas correctamente.

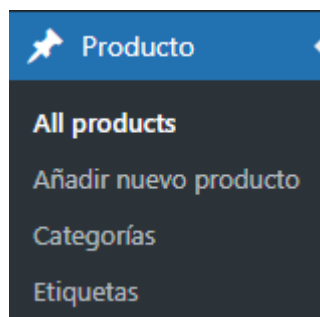


Figura 23. Funcionalidad en el menú de WordPress

Add new Taxonomy

Nombre

El nombre es cómo aparecerá en tu sitio.

Slug

El «slug» es la versión amigable de la URL para el nombre. Suele estar en minúsculas y contiene solo letras, números y guiones.

Taxonomy parent

Asigna un término superior para crear una jerarquía. El término jazz, por ejemplo, sería el superior de bebop y big band.

Descripción

La descripción no suele mostrarse por defecto, sin embargo hay algunos temas que puede que la muestren.

[Add new Taxonomy](#)

Figura 24. Ejemplo de creación de categoría.

2. Se crea una categoría nueva y se le añade al menos un subapartado. En este caso se añade la subcategoría pantalón a la categoría ya creada ropa como se muestra en la figura 24. Y se comprueba que se ha creado correctamente, véase figura 25.

| | | | | |
|--------------------------|------------------------------|-----------------------|------------|---|
| <input type="checkbox"/> | ropa | — | ropa | 2 |
| <input type="checkbox"/> | — Pantalones | Tipo de ropa pantalón | pantalones | 0 |

Figura 25. Resultado de creación de categoría pantalón.

3. Se crea una etiqueta nueva y se le añade al menos un subapartado. El ejemplo de creación y resultado es el mismo obtenido que con las categorías, con la diferencia de que el de las etiquetas sucede en la página de “Etiquetas” en el menú de la figura 23.
4. Con ambas taxonomías probadas es necesario crear un producto en el catálogo, asignarlo a grupos de tanto categorías como etiquetas y comprobar que todo se guarda correctamente.

zapatilla roja

es una zapatilla de color rojo

Categorías

☒ ropa

☒ calzado

☐ Pantalones

☐ Sin categorizar

[Add new Taxonomy](#)

Etiquetas

☒ Montaña

[Add new Taxonomy](#)

Figura 26. Creación de un producto y valores de las taxonomías asignadas.

Se comprueba el correcto guardado accediendo a la edición del producto mostrado y verificando que los valores introducidos en la creación no han sido alterados.

Pruebas de la página de configuración CMB2

Para probar la página de configuración de CMB2, se rellenan en los campos de la página nuevos datos para comprobar si todos los datos se sobrescriben correctamente.

Inicialmente los nuevos valores introducidos en la página de configuración no quedaban sobrescritos correctamente. Esto se debe a que los hooks que llaman a las funciones que crean el Custom Post Type y sus dos taxonomías eran llamados durante el evento 'init', y este evento sucede antes de que se carguen las funcionalidades del plugin CMB2, por lo que nunca se registraban los nuevos valores. En base a esto, se cambia el evento que activa estos hooks por un evento propio que introduce en el sistema el plugin CMB2, "cmb2_init". Este nuevo evento lo ejecuta el plugin CMB2 mismo una vez se ha cargado por completo, por lo que en ese momento la página de configuración ha sido creada y carga correctamente los datos.

| | | |
|---------------------------------------|---|---|
| Editar slug de los productos | <input type="text" value="products"/> | Modifica el valor del producto en la url |
| Editar slug de la taxonomía categoría | <input type="text" value="categories"/> | Modifica el valor de la taxonomía de categoría del producto en la url |
| Editar slug de la taxonomía etiqueta | <input type="text" value="tags"/> | Modifica el valor de la taxonomía de la etiqueta del producto en la url |
| Editar el nombre del producto | <input type="text" value="Bicicleta"/> | Modifica el nombre que se muestra publicamente del producto |

Figura 27. Página de configuración.

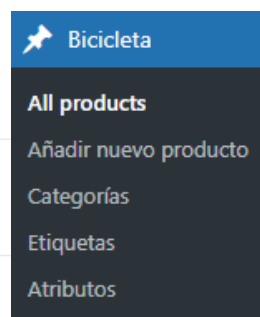


Figura 28. Funcionalidad en el menú de WordPress cambiada.

Pruebas del tema

Se generan varios productos en el catálogo para comprobar que el tema muestra correctamente su información. Además, se comprueba que la transición entre pantallas sea correcta y que los slug definidos para cada apartado correspondan con las URL del sitio.



Figura 29. Página de productos existentes.

La simplicidad del tema estético es un requerimiento del proyecto, ya que su finalidad es comprobar la que la estructura de los productos se ha implementado correctamente. Debe servir de punto de referencia para implementar futuros catálogos para clientes.

Pruebas del endpoint

Se accede desde el navegador a la ruta:

http://ttandem.com.devel/wordpress/wp-json/*/products/categories/25

Donde 25 es el id de la categoría de la que se quieren recibir los productos y el asterisco "*" corresponde a un pequeño fragmento de la URL que se prefiere mantener privada.



Figura 30 . Página de un producto.

```
[
  {
    "ID": 15,
    "post_author": "1",
    "post_date": "2021-05-27 12:41:09",
    "post_date_gmt": "2021-05-27 12:41:09",
    "post_content": "<!-- wp:paragraph -->\n<p>Zapatilla deportiva de color rojo, material algod\u00f3n...</p>\n<!-- \\/wp:paragraph -->",
    "post_title": "Zapatilla roja",
    "post_excerpt": "",
    "post_status": "publish",
    "comment_status": "closed",
    "ping_status": "closed",
    "post_password": "",
    "post_name": "zapatilla-roja",
    "to_ping": "",
    "pinged": "",
    "post_modified": "2021-05-27 12:41:09",
    "post_modified_gmt": "2021-05-27 12:41:09",
    "post_content_filtered": "",
    "post_parent": 0,
    "guid": "http:\\\\ttandem.com.devel\\wordpress\\?post_type=product&#038;p=15",
    "menu_order": 0,
    "post_type": "product",
    "post_mime_type": "",
    "comment_count": "0",
    "filter": "raw"
  },
]
```

Figura 31. Captura de parte de la respuesta JSON de la api.

En este caso la categoría 25 es la categoría de calzado, y en la imagen se verifica que el primer producto devuelto es de una “zapatilla de color rojo”. Se confirma que en los datos obtenidos solo aparecen los productos del catálogo con el id de la categoría pedida por la URL definida en el endpoint.

Pruebas de funcionalidad de atributos

Con los contenidos programados que se han explicado en la fase de 5 desarrollo “Atributos de Woocommerce”, se debe comprobar:

- Que la página de creación de un nuevo atributo almacena correctamente el valor en la base de datos.

| | | | | | | | |
|--|--------------------------|--|--|--|---|---------------|----------------|
| Nombre | <input type="checkbox"/> | | | | 1 | color | Color |
| Talla calzado | <input type="checkbox"/> | | | | 2 | tallapantalon | Talla pantalón |
| Nombre para el atributo (mostrado en la tienda). | <input type="checkbox"/> | | | | 3 | tallacalzado | Talla calzado |
| Slug | | | | | | | |
| tallacalzado | | | | | | | |
| Slug/referencia única del atributo; debe tener menos de 28 caracteres. | | | | | | | |

Figura 32. Datos de creación de nuevo atributo.

Figura 33. Tabla de atributos con nuevos datos

- Que la función creada para leer la tabla de atributos de la base de datos y la que genera nuevas taxonomías en base a ello funciona correctamente. Al igual que permite crear subgrupos dentro de un atributo y los relaciona correctamente dentro de cada taxonomía.

| | | | |
|----------------|---------------|---------------------|--|
| Color | color | Orden personalizado | Rojo, Verde Configurar términos |
| Talla calzado | tallacalzado | Orden personalizado | 42, 43, 44 Configurar términos |
| Talla pantalón | tallapantalon | Orden personalizado | 33, 43, 44 Configurar términos |

Figura 34. Datos del nuevo atributo en WordPress.

- Que WordPress permite asignar valores de las taxonomías de atributos creados a los productos, al igual que se muestra en la figura 26 con las categorías y etiquetas.

Esto no era posible de efectuar. Uno de los contratiempos mencionados durante el desarrollo de los atributos es que Woocommerce les da una estructura similar a las taxonomías de WordPress. Esto causa que WordPress no muestre los atributos durante la creación o edición de un producto para relacionarlos entre sí.

Para solucionar este percance se genera una función en el plugin, que crea un metabox de selección por cada atributo existente en las páginas de modificación del producto. Se genera la estructura de estos metabox y los botones mediante HTML y se colocan debajo de los campos de selección que ya existen de categorías y etiquetas.

The image shows two parts of a WordPress product creation interface. On the left, a metabox titled 'product_cat' is expanded, showing a list of categories: 'ropa', 'calzado', 'Pantalones', and 'Sin categorizar', each with an unchecked checkbox. Below this list is a link 'Add new Taxonomy'. Below the 'product_cat' section are four other attribute sections: 'product_tag', 'Color del producto', 'Talla calzado del producto', and 'Talla pantalón del producto', each with a dropdown arrow. On the right, the 'Talla calzado del producto' attribute is expanded, showing three size options: '42', '43', and '44', each with an unchecked checkbox.

Figura 36. Opciones del atributo desplegadas.

Figura 35. Metabox de los atributos en la página de creación del producto.

Tras lograr que todas estas comprobaciones arrojen resultados satisfactorios, se ha verificado que los atributos de Woocommerce funcionan correctamente y se comportan como una taxonomía.

Además, hay que comprobar que todo lo creado sigue siendo reconocible por Woocommerce, siendo este el objetivo principal del proyecto. Por lo que desactivando el plugin del catálogo y reactivando el de Woocommerce se confirma que los nuevos atributos creados y relaciones de los productos también se han guardado.

- ¿Qué sucede si se eliminan las tablas creadas por Woocommerce de la base de datos y se intenta crear un atributo desde el catálogo?

El funcionamiento de WordPress se detiene y hay que reiniciar el plugin. Hasta el momento se han realizado todas las pruebas y ejecuciones partiendo de que la tabla de los atributos ya había sido creada. Esto sucede automáticamente cuando se activa Woocommerce, pero no se debe depender de haber trabajado anteriormente con este plugin para que el catálogo funcione de forma correcta.

Por lo tanto, se debe crear una función que cree la tabla en la base de datos en caso de que esta no exista cuando el plugin de TTANDEM se activa.


```

global $wpdb;

$charset_collate = $wpdb->get_charset_collate();

$sql = "CREATE TABLE $table_name (
    id mediumint(9) NOT NULL AUTO_INCREMENT,
    time datetime DEFAULT '0000-00-00 00:00:00' NOT NULL,
    name tinytext NOT NULL,
    text text NOT NULL,
    url varchar(55) DEFAULT '' NOT NULL,
    PRIMARY KEY (id)
) $charset_collate;";

require_once( ABSPATH . 'wp-admin/includes/upgrade.php' );
dbDelta( $sql );

```

Figura 37. Ejemplo de creación de una tabla en la base de datos de WordPress. [Codex de WordPress](#) [4].

```

register_activation_hook(__FILE__, 'check_attribute_table');

```

Figura 38. Hook de llamada de la función que crea la tabla en el plugin.

En este ejemplo del codex de WordPress se muestra una función similar a la generada para el producto. El hook de la figura 38 llama a la función que crea la tabla en la base de datos cada vez que el plugin de TTANDEM se activa. La variable global `$wpdb` se encarga de recibir el script en SQL. Y se llama mediante la función `dbDelta()`, una función de actualización de la base de datos que se ocupa de comprobar si la tabla que se quiere crear existe, y en caso de no ser así, la crea.

Conclusiones

Después de todas las pruebas realizadas, el catálogo de TTANDEM es totalmente funcional y proporciona las siguientes funcionalidades:

1. Permite generar un catálogo de productos online.
2. Incorpora la agrupación de productos en base a los tres índices de agrupación más usados en la tiendas online: categorías de productos, etiquetas y atributos de productos.
3. El plugin es una plantilla que servirá de punto de partida para generar proyectos personalizados para clientes, por lo tanto no acota ningún apartado ni se desaprovecharán fragmentos de código.
4. Proporciona la posibilidad de modificar los campos que ha sido necesario programar pero que deben ser personalizables para proyectos futuros. Esto evita que se altere el código.
5. El catálogo online dispone de una ruta de salida en la API de WordPress que permite enviar datos sobre los productos existentes por categorías.
6. Todas los objetos creados en el catálogo son totalmente compatibles con Woocommerce. Por lo que, si hay necesidad de crear una tienda online a partir del catálogo, la transferencia de todos los datos entre ambos plugins se produce de forma totalmente automática.

Se realiza una comparación entre lo que cuesta aproximadamente traspasar los datos de un catálogo a una tienda nueva de Woocommerce de forma manual a TTANDEM frente a lo que se tarda usando el catálogo de productos creado.

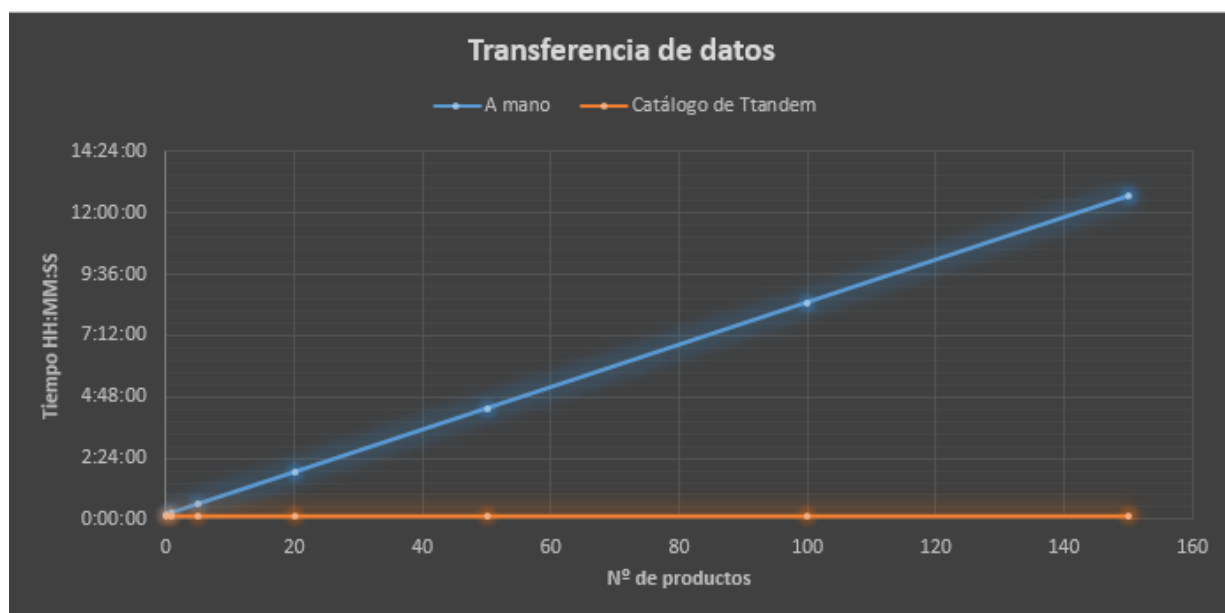


Diagrama 3. Comparación de tiempos en el traspaso de datos.

La diferencia de tiempo entre ambos procesos es considerable. Mientras que el tiempo necesario para copiar información con el método tradicional aumenta respecto al número de productos existente, el producto creado tarda aproximadamente 5 minutos en pasar todos los datos, independientemente del volumen del catálogo existente.

De media, un catálogo de productos de los clientes de TTANDEM ronda entre los 80-100 productos, aunque los ha habido mucho mayores. Por lo que el uso de esta herramienta ahorra más de 6 horas de trabajo como mínimo cada vez que se quiera convertir un catálogo a tienda. Cabe mencionar que como el catálogo automatiza el traspaso de información se eliminan los errores humanos, que pueden tener repercusiones económicas graves.

Se ofrece un documento (ANEXO 1) como guía de uso general del plugin de catálogo de productos de TTANDEM. En dicho anexo se pueden apreciar las funcionalidades resultantes del proyecto.

Cierre del proyecto

El proyecto planteado por TTANDEM ha resultado ser muy complejo pero interesante de desarrollar. Gran parte del ciclo de vida del proyecto ha transcurrido investigando y analizando en pos de definir qué era posible desarrollar y qué no. Esto se debe a que actualmente no existe un producto que ofrezca un servicio similar en el mercado. Además, en su comienzo, el proyecto ponía en duda la viabilidad de ciertos aspectos, como la creación de atributos, que finalmente han podido ser implementados satisfactoriamente.

El producto final proporciona a la empresa TTANDEM un producto base que contiene la estructura para generar catálogos totalmente personalizados a medida de los clientes.

Adicionalmente, la finalización del producto ha servido para demostrar que es posible una generación de un catálogo compatible con Woocommerce. Por lo tanto, el producto abre vías para una nueva rama de productos para la empresa.

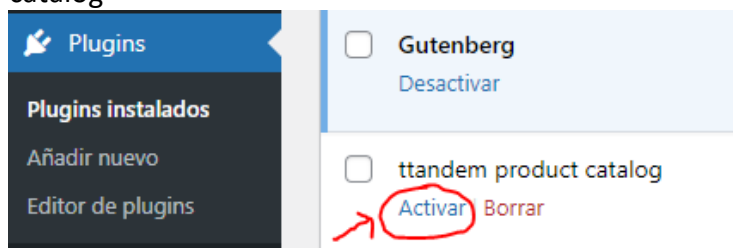
Bibliografía

- [1] Página del codex de WordPress sobre la creación de hooks:
<https://developer.wordpress.org/plugins/hooks/actions/>
- [2] Ejemplo de creación de Custom Post Type y taxonomías:
<https://www.dariobf.com/custom-post-types-wordpress/>
- [3] Kinsta blog de Brian Jackson sobre el editor Gutenberg:
<https://kinsta.com/es/blog/gutenberg-wordpress-editor/>
- [4] Página del codex de WordPress sobre creación de tablas en la base de datos mediante plugins.
https://codex.wordpress.org/Creating_Tables_with_Plugins

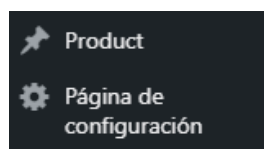
ANEXO 1

Instalación del plugin

1. Se copia la carpeta “ttandem-product-catalog” que contiene el plugin al directorio de WordPress /wp-content/plugins/
2. En la pestaña “Plugins” de WordPress se activa el plugin “ttandem product catalog”

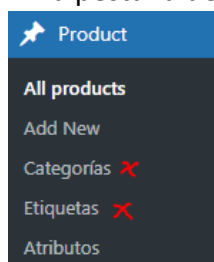


3. Se añadirán las pestañas de “Product” y “Página de configuración” automáticamente al menú de WordPress.



Creación de una categoría o Etiqueta

1. En la pestaña de “Product” se selecciona “Categorías” o “Etiquetas”.



2. Rellena los campos con la información deseada y pulsa el botón de añadir.

Nombre

El nombre es cómo aparecerá en tu sitio.

Slug

El «slug» es la versión amigable de la URL para el nombre. Suele estar en minúsculas y contiene solo letras, números y guiones.

Taxonomy parent

Asigna un término superior para crear una jerarquía. El término jazz, por ejemplo, sería el superior de bebop y big band.

Descripción

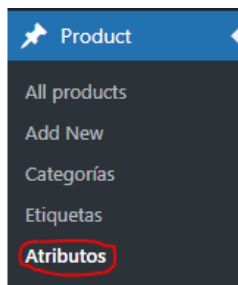
La descripción no suele mostrarse por defecto, sin embargo hay algunos temas que puede que la muestren.

[Add new Taxonomy](#)

3. (Opcional) En caso de que la taxonomía que quieras añadir sea un subgrupo de una creada selecciona en la pestaña “Taxonomy parent” la taxonomía padre o superior.

Creación de un producto

1. En la pestaña de “Product” se selecciona “Atributos”.



2. Rellena los dos campos de “Nombre” y “Slug” y añade el atributo.

Añadir nuevo atributo

Los atributos te permiten definir datos adicionales del producto, como talla o color. Puedes usar atributos en la barra lateral usando los widgets de «navegación por capas».

Nombre

Nombre para el atributo (mostrado en la tienda).

Slug

Slug/referencia única del atributo; debe tener menos de 28 caracteres.

☐ ¿Activamos el archivo?
 Activa esto si quieres que este atributo tenga páginas de archivo de producto en tu tienda.

Orden de clasificación por defecto

Determina el orden de los términos en las páginas de productos de la tienda. Si usas un orden personalizado puedes arrastrar y soltar los términos en este atributo.

Añadir atributo

- Para crear subgrupos dentro de los atributos en el recuadro de la derecha pulsa en un atributo existente.

| Nombre | Slug | Ordenar por | Términos |
|---------------|--------------|---------------------|--|
| Color | color | Orden personalizado | Azul, Rojo, Verde Configurar términos |
| Material | material | Orden personalizado | Algodón, Seda Configurar términos |
| Talla Calzado | tallacalzado | Orden personalizado | 37, 38, 39, 40 Configurar términos |

- Añade los datos como si de una taxonomía se tratase y pulsa añadir.

Añadir nuevo Color

Nombre

El nombre es cómo aparecerá en tu sitio.

Slug

El «slug» es la versión amigable de la URL para el nombre. Suele estar en minúsculas y contiene solo letras, números y guiones.

Padre Color

Asigna un término superior para crear una jerarquía. El término jazz, por ejemplo, sería el superior de bebop y big band.

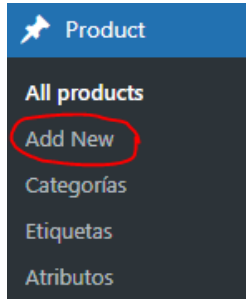
Descripción

La descripción no suele mostrarse por defecto, sin embargo hay algunos temas que puede que la muestren.

Añadir nuevo Color

Creación de un producto

1. En la pestaña de “Product” se selecciona “Add New”.



2. Rellena los campos de “Título” y “Descripción”.

Título |

Teclea / para elegir un bloque

3. Relaciona el producto con taxonomías existentes.

Categorías

- ☐ Comida
- ☒ Ropa
 - ☐ Calzado
 - ☐ Calzado deportivo
 - ☒ Sandalias
 - ☐ Sudaderas

Etiquetas

- ☐ Deporte
- ☐ Montaña
- ☒ Verano

Color del producto

- ☒ Azul
- ☐ Rojo
- ☐ Verde

4. Añade una imagen.

Imagen destacada

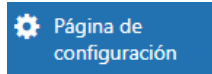


Establecer la imagen destacada

5. Publica el producto.

Modificación usando la página de configuración CMB2

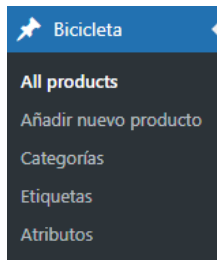
1. Selecciona “Página de configuración” en el menú de WordPress.



2. Rellena los campos con los nuevos parámetros que desees y pulsa guardar.

| | | |
|---------------------------------------|---|---|
| Editar slug de los productos | <input type="text" value="products"/> | Modifica el valor del producto en la url |
| Editar slug de la taxonomía categoría | <input type="text" value="categories"/> | Modifica el valor de la taxonomía de categoría del producto en la url |
| Editar slug de la taxonomía etiqueta | <input type="text" value="tags"/> | Modifica el valor de la taxonomía de la etiqueta del producto en la url |
| Editar el nombre del producto | <input type="text" value="Bicicleta"/> | Modifica el nombre que se muestra públicamente del producto |

3. Se actualizarán automáticamente los campos.

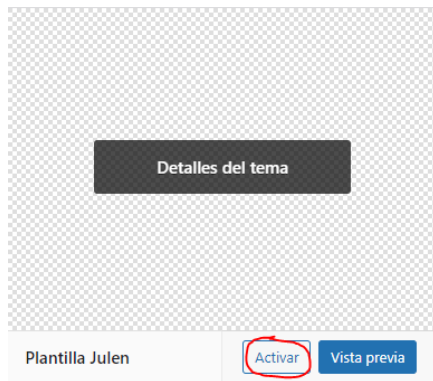


Instalación del tema

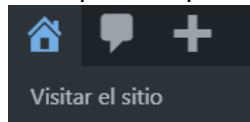
1. Se copia la carpeta “plantilla-julen” que contiene el tema al directorio de WordPress /wp-content/themes/
2. Se accede a la pestaña “Apariencia” en el menú de WordPress.



3. Activa el tema.



4. En la parte superior de WordPress se selecciona visitar el sitio.



5. Con esto se accede a la visualización de los productos. No es necesario activar el tema “plantilla-julen”, se puede visualizar el catálogo con cualquier plantilla de WordPress.



ANEXO 2

Introducción

En este documento se realizará un análisis de las páginas web: virto productos, snovit cadenas de nieve, findus food services productos congelados y hoteleria e isimar sillas y sillones de diseño exterior. Dicho análisis tiene como fin encontrar componentes y requisitos comunes entre las distintas webs para aplicarlos en el proyecto de wordpress.

Página principal

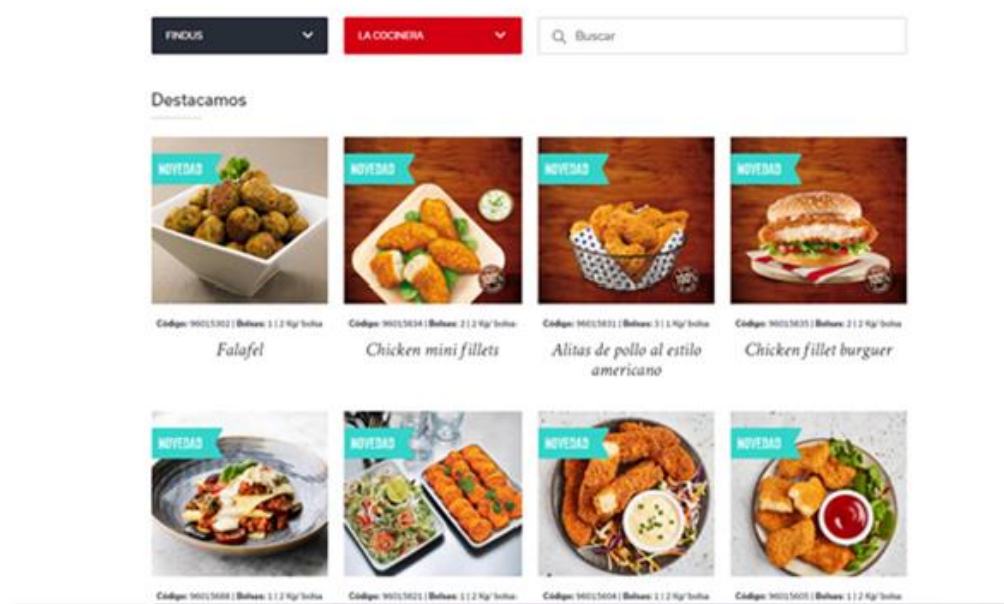
Apartado informativo o comercial

Todas las páginas antes de empezar a mostrar los productos muestran un apartado (de tamaño variable entre las distintas webs) donde se presentan a ellos mismo con una descripción de los productos o el lema de la empresa.

Área de productos /catálogo

Es la parte principal de la página principal, está compuesto por un buscador o filtro y luego muestra los productos.

Buscador y filtro: El buscador es simplemente un área de texto donde introducir un producto o característica y que al buscar los productos que aparezcan tengan algo en común con lo buscado. El o los filtros veo recomendable hacer uno o dos campos desplegables con categorías o características de los productos que ofrece la empresa.



Productos: La forma más usada y personalmente pienso que visualmente la más bonita también es mostrar los productos en un formato de imágenes cuadradas, estilo celdas. Con una imagen de tamaño mediano/pequeño que muestre un producto y debajo de esta el nombre y algunos datos como precio.

Lo más repetido a la hora de que productos se muestran es separar los productos por tipos/apartados, y mostrar en la página principal solo los más destacados de cada tipo, y habilitando un botón que lleve a una página donde se vayan a mostrar todos los productos de ese tipo concreto si se desea.

Página de tipo de producto

Es una extensión del catálogo, por lo que he podido analizar que se puede crear esta página a parte si en la página principal solo se han mostrado productos destacados de cada tipo. El querer ver todos los productos ya sea usando el filtro de la página principal o simplemente clicando en “ver más” de un tipo o en nombre del tipo de producto concreto, se accede a esta página.

Visualmente es idéntica a la página principal, solo se distingue en que puede mostrar información sobre el tipo de producto específico y que muestra todos los productos del tipo, no sólo los relevantes.



Página de un producto

Al clicar en un producto se abre una página con información detallada del producto. Las características que más se repiten entre las distintas webs son las siguientes:

Título con el nombre del producto en grande.

Imagen/es: La imagen que se mostraba en el catálogo, pero con un tamaño mayor. Dependiendo del producto que se desea vender veo recomendable añadir fotos extra para dar más información. Por ejemplo, con ropa o muebles.


Descripción o características: Es necesario añadir una descripción detallada del producto. En cada tipo de producto será distinta, pero veo necesarias características físicas como dimensiones e instrucciones de uso en la mayoría de los tipos de productos. En el caso de la comida las instrucciones de uso son formas de preparación.


Chicken mini fillets

CÓDIGO: 96015834 | BOLSAS: 2 | 1.5 Kg/ BOLSA

Inicio / Productos / Chicken mini fillets

Tiemas tras de pechuga marinadas y empanadas cubiertas con un delicioso rebocado crujiente que no dejará a nadie indiferente.

**Horno:** Precalentar a 200°C.

**Freír:** Precalentar a 175°C.

Ingredientes

Pechuga de pollo 60%, pan rallado (harina de TRIGO), sal, levadura, pimentón, cúrcuma, agua, aceite de rubia, harina de TRIGO, semillas de SESAMO 2%, dextrosa, sal, extracto de pimienta. Puede contener: HUEVO, SOJA, MOSTAZA, AJO, LECHE.

Reconstitución


HORNO

1. Precalentar el horno a 200°C.
2. Colocar las piezas sobre papel de hornear en la bandeja a media altura y cocinar durante 15 minutos.
3. Dar la vuelta a las piezas pasados 7-8 minutos para que el producto se dore uniformemente.

FREIDORA

1. Precalentar el aceite a 175°C.
2. Freír las piezas durante 5 minutos.

Datos logísticos



NOVEDAD

100% Proteína

Alérgenos: TRIGO, SESAMO, HUEVO, SOJA, MOSTAZA, AJO, LECHE.
Ver listado completo de alérgenos

Compartir producto

Conclusión

Ciertos componentes de la web dependerán de los productos y servicios que quiera ofrecer la empresa del cliente. Al igual que el encabezado, pie y estética general de la página dependerán también del cliente a no ser que se especifique lo contrario en su contrato.

El catálogo, que sería la parte fundamental, debe ser claro y de uso intuitivo. El hecho de filtrar los productos en la página principal o separarlos por páginas de tipo es secundario y simplemente estético. Pero un producto en el catálogo debe diferenciarse claramente con la imagen y descripción proporcionadas. En cuanto a la página del producto debe ser detallada y muy completa.

Enlaces a las tiendas analizadas

- 1) [Virto productos](#)
- 2) [Snovit cadenas de nieve](#)
- 3) [Findus food service productos congelados y hosteleria](#)
- 4) [Isimar sillas y sillones de diseño exterior](#)